

## CSCD01 Deliverable 2

Team 34: 3+3

### **Issues & Descriptions**

#### **Issue #661**: "Other amount" text field allows amounts in scientific notation

**Description:** When the user selects "Other amount" on the front page's donation form, the form accepts scientific notation as a valid input and converts it into an integer. For example, entering the value "1e3" will be parsed as a \$1000 donation. Interestingly, the input field rejects any characters except for the "e" character, which allows the scientific notation bug in the first place.

Amounts in scientific notation do not affect the hard-coded donation limit of \$10,000,000., so if "1e12" is inputted, a popup notifies the user that the value must be less than or equal to \$10,000,000.

**Estimated amount of work: 2 hours**

#### **Issue #691**: Add "no value selected" error state on landing home page

**Description:** When landing on the homepage, there are two payment methods that Mozilla provides to its donors. When users simply click on the Visa payment button, the homepage refresh itself without displaying any visible error prompts, also choose another currency and repeat the actions above will route back to default currency that is (USD).

We could set a default value at the homepage or add error message prompts to avoid reloading the page which causes unnecessary inconveniences. Function `showErrorMessage()` in `main.js` could use some extra triggers.

**Estimated amount of work: 1 hour**

#### **Issue #642**: Set a better default country

**Description:** When the users/donors open the donate site, the default country is always set to the US and the language is English, also the currency is USD. Which is not convenient for the users in other countries or the non-English speaking users.

We could use the variable 'buyers-IP' which is already in the code and pass it around so we can set a better default country according to the 'buyers-IP'. Another option is to use an API([ipdata](#)) to get the users' IP, language, and currency but this seems unnecessary since we can make use of the existing variable.

**Estimated amount of work: 2 hours**

**Issue #761:** Proper error message when clicking the PayPal button

**Description:** When the user clicks the PayPal button without selecting an amount to donate for, the page shows an error message saying “There was an error processing your payment. Please try again.”, which gives no context to the user what the actual problem is, and if the user listens and keeps trying again, he will be stuck there seeing this error message repeatedly and not knowing that he should pick an amount to donate for.

**Estimated amount of work: 2 hours**

**Issue #484:** Possibility of accidentally donating one-time instead of monthly

**Description:** When you select a monthly donation option and choose visa payment, once you press the back button to the previous page the donation form is back to a one-time payment option but the radio input button is set to monthly as opposed to one-time.

Once you go back to the previous page, neither of the ‘one-time’ or ‘monthly’ labels are set to active. The solution can be to set the tab panels to be set to whatever the radiolabel was initially set to or make the radio label reset back to one-time.

**Estimated amount of work: 1 hour**

**Issue #367:** Zip code has no error for Invalid Input:

When given a Zip/Postal code for payment method the field raises no errors what an input that does not match the Zip/Postal code pattern. This allows users to make potential errors in the form costing that might make them leave the site. This costs Mozilla money.

We chose not to do this issue because it could cause other issues trying to adjust for countries that have different Zip/Postal code formats.

**Estimated amount of work: 3 hours**

## **Bug Fixes - Technical Commentary**

We decided to fix two bugs from our list: Issue #642 and Issue #484.

We chose #642 because we thought that it had a high impact to users because when users are donating, there should be as little friction as possible during the

payment process in order to not deter them from cancelling their donation. Issue #642 reduces the friction in the payment process by setting the payment form's default country automatically, which means the user fills in fewer fields. Technically, it was also a good challenge since it would involve diving into how Django handles user locales, countries and languages, which is usually a complicated process.

We chose #484 because we thought that the problem might have a big impact on users, for example, when used for some reason decides to go back to the main page when they arrived the visa payment page with selecting monthly payment, the radio button appears as monthly yet the form itself is treated as if the user selected one-time, and now if the user clicks visa payment with selected donation amount, he/she might still think that he/she is paying a monthly donation yet it is actually a one-time donation. We also decided on this one not only because it would help a lot of potential users, but also because it is also not an easy issue to solve as some other issues that we have chosen, it is nice to have some challenges in the way. We have to understand the structure of their JS files as well as knowing the interaction between the JS file and the HTML components to be able to solve this issue.

#### **Issue #642: Set a better default country**

Rather than always setting the country dropdown in the default form to USA for all locales we now set the default country based on the user's IP.

The country code is determined using Django's `get_language()` function which is based on the user's IP address. Then by using the `LOCALE_COUNTRY_MAP` added to `constants.py`, the default country is determined with the country code.

Working branch:

- `set-default-country`

Files modified:

- `donate/payments/constants.py`
- `donate/payments/forms.py`

Changes are shown in the following commit:

<https://github.com/CSCD01/donate-wagtail-team34/commit/e27af3731b0227cc7a6b524b4ae21e8dec448ab2>

#### **Issue #484: Possibility of accidentally donating one-time instead of monthly**

The donation frequency is determined using the `LocalStorage` variable 'monthly'. It changes when the users click on the radio buttons to change the frequency (i.e. the value of 'monthly' is set to false by default and once the click event is triggered on the radio buttons, if the target frequency option button is monthly we then change the `LocalStorage` variable to true). We are also using another `localStorage` variable "single" to determine whether the user has clicked one-time after clicking monthly,

and if the user has clicked one-time after monthly, the monthly localStorage variable will be back to False when the user leaves the landing page indicating that he selected one-time payment. When the main.js finishes loading the DOM contents, if 'monthly' is true we then remove the HTML class "tabs\_panel--hidden" in 'the tab-panel-2' element which changes the CSS style to display the tab-panel for the monthly options as well as having the 'monthly' radio button as the selected one. Correspondingly, we also need to hide the corresponding one-time form, which we added tabs\_panel--hidden to the tab-panel-1 element in order to only show one correct form at a time. When no changes are made (i.e. the user did not select the Monthly radio button before), the one-time tab is displayed.

Working branch:

- dev/donation\_option\_fixes

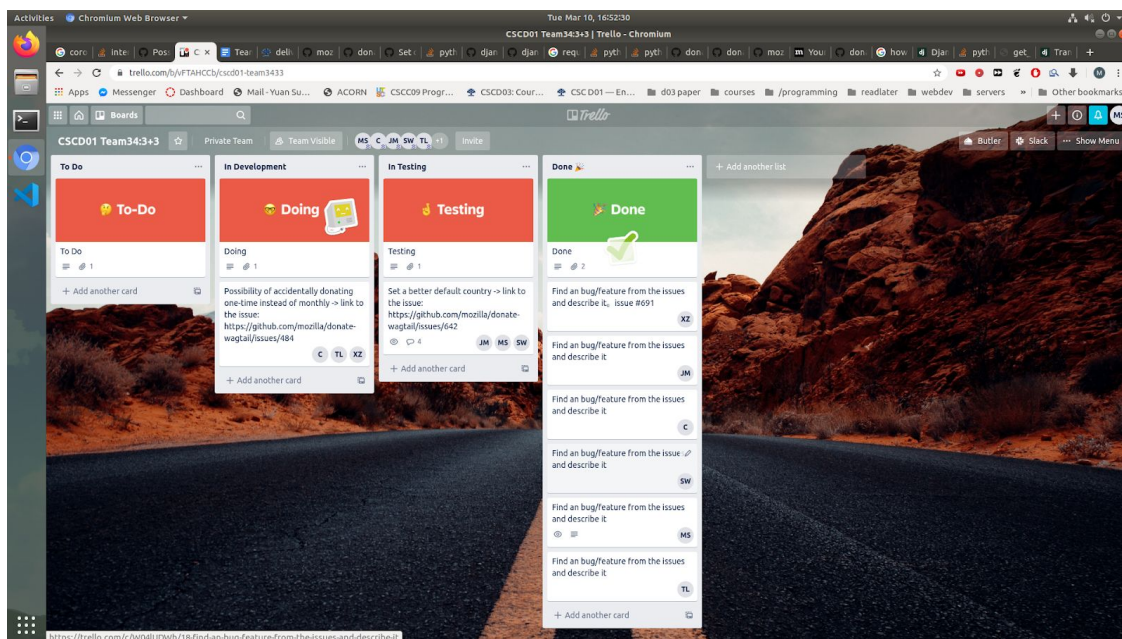
Files modified:

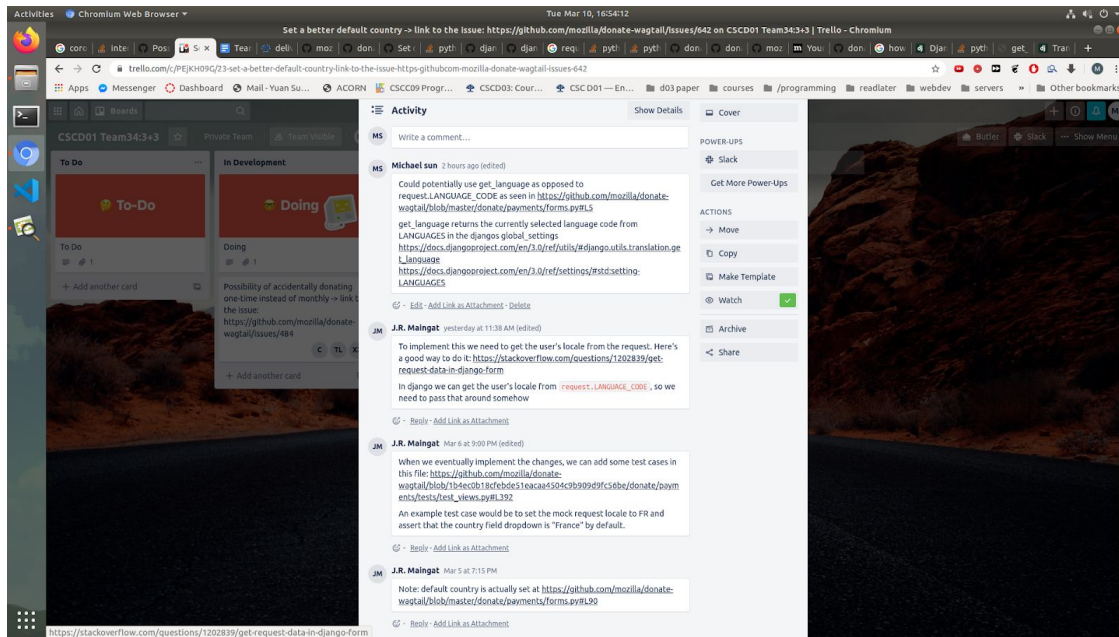
- source/js/main.js
- source/js/components/tabs.js

Changes are shown in the following commit:

<https://github.com/CSCD01/donate-wagtail-team34/commit/2b2d946072016a2e83c1c2d42302522dee8de12>

## Software Development Process





Here are a couple of screenshots of our Kanban board during our software development process. The front of the cards on the board displays the tasks and team members assigned to, while the back of the card (clicking on it) displays more detailed information about documentation and resources.

We strictly followed the WIP that maximum tasks for this deliverable did not exceed 6. The two cards for bugs that we choose to implement are pulled into the development stage after the 6 cards are done.

Before the implementation, our group had a stand-up to pick issues together and vote for the top 2 bugs that we are all comfortable to implement to ensure everybody is clear with the common goals.

We divided the six-person team into two three-person teams, each team responsible for the implementation and description and test suite of a bug to make sure we can manage the smoothness of the workflow.

## Acceptance Test Suites

### Issue #484: Possibility of accidentally donating one-time instead of monthly Test Suites:

The idea of the tests is using the elements' clickability to check their visibility since the hidden elements are not clickable and will raise an exception. If there are no exceptions raised when performing the click action on the desired element, then the element exists and the test passes. And we noticed that when no amount is selected users will be redirected back to the default page, so we added two test

cases where the donation frequency is selected but no amount selected/entered. And we also added two test cases where the target site is French to make sure the implementation works not only for the default (i.e. English) sites but for all other languages.

**test\_one\_time.py**(corresponding to scenario #1)

**test\_monthly.py**(corresponding to scenarios #2)

- The idea here is to perform the click actions normal users would make and see if the tabs are displayed correctly once users hit the go back button in the browser.

**test\_one\_time\_with\_on\_amount.py**

**test\_monthly\_with\_no\_amount.py**

- The idea here is to check the visibility of the correct tab-panel when the donation frequency is selected but no amount is selected
- Note: when there is no amount selected, it redirects the user to the default page (i.e. 0.0.0.0:8000). This can easily happen if users accidentally click on the VISA payment button with no amount selected/entered.

**test\_one\_time\_in\_fr.py**

**test\_monthly\_in\_fr.py**

- The target site in these two tests is 0.0.0.0:8000/fr/.
- The idea here is check if the fix works for other languages.

#### **Test files:**

[https://github.com/CSCD01/donate-wagtail-team34/tree/dev/donation\\_option\\_fixes/donate/payments/tests](https://github.com/CSCD01/donate-wagtail-team34/tree/dev/donation_option_fixes/donate/payments/tests)

- donate/payments/tests/test\_monthly.py
- donate/payments/tests/test\_monthly\_in\_fr.py
- donate/payments/tests/test\_monthly\_with\_no\_amount.py
- donate/payments/tests/test\_one\_time.py
- donate/payments/tests/test\_one\_time\_in\_fr.py
- donate/payments/tests/test\_one\_time\_with\_no\_amount.py

Tool: python2.7+, Selenium webdriver.

#### **Usage:**

1. open Terminal, input: pip3 install -U selenium.
2. cd to the directory ../donate-wagtail-team34/donate/payments/tests.
3. run "python3 test\_monthly.py" to test the monthly option or "python3 test\_othertests.py" for running other tests by replacing the 'othertests' with one of the options in the list -> [ test\_one\_time, test\_one\_time\_in\_fr, test\_one\_time\_with\_no\_amount, test\_monthly\_in\_fr, test\_monthly\_with\_no\_amount ]



4. verify and validate the results.

**Issue #642: Set a better default country**

For the UAT of #642, there is one user scenario for each supported locale that follows this format.

Users from [Locale] navigate to the detail form, the country drop down on this page is already set to the correct Locale. The acceptance tests should verify that this is true for each accepted Locale

**test\_views.py:** Tests run through all the accepted locales and check that the drop box default is changed as expected

**Test files:**

donate/payments/tests/**test\_views.py**

Tool: python2.7+, Selenium webdriver.

**Usage:**

5. open Terminal, input: pip3 install -U selenium.
6. cd to the directory `../donate-wagtail-team34/donate/payments/tests`.
7. run `"python3 test_views.py"` to test the views for the app including the locale fix tests
8. verify and validate the results.